

Amendments to the Claims

Applicant respectfully requests reconsideration of this application as amended. Claims 1-46 have been canceled. Claims 47-70 have been added.

Listing of Claims:

Please cancel claim 1 without prejudice.

2-46 (canceled)

47. A method comprising:
 - applying a first method from a first execution object, said first execution object identifying a first metadata object corresponding to a first function, said first function requiring one or more input parameters, said first metadata object storing data describing each input parameter of said first function, said first method causing the acts of,
 - accessing the data describing each input parameter of said first function from said first metadata object;
 - attempting to match values associated with said first execution object to each input parameter of said first function as described by the data; and
 - determining a value is missing for at least a first input parameter to said first function.
48. The method of claim 47, wherein said first method further causes the acts of:
 - locating a second metadata object corresponding to a second function having one or more output parameters, said second metadata object storing data describing each output parameter of said second function;

determining the missing first input parameter is an output parameter of said second function; and
executing said second function to acquire the missing value.

49. The method of claim 48, wherein said first method further causes the act of:
associating the acquired value with said first execution object; and
executing said first function using the acquired value now associated with said first execution object as the first input parameter.
50. The method of claim 47, wherein said first method further causes the acts of:
determining a set of metadata objects that each have stored therein data describing an output parameter that matches one or more of the missing input parameters, wherein each metadata object in said set corresponds to a different function having a set of one or more output parameters, each metadata object in said set storing data describing said set of output parameters for the corresponding function; and
executing the functions corresponding to the set of metadata objects to acquire said set of missing values;
associating the acquired values with the first execution object; and
executing said first function using the values associated with the first execution object as input parameters.
51. The method of claim 47, wherein said attempting further includes:
accessing a structure in said first execution object, said structure in said first execution object to store values for one or more said set of input parameters to said first function.
52. The method of claim 47, wherein said attempting further includes:

accessing a structure in a manager object identified by a structure in said first execution object, said structure in said manager object identifying a default one of a plurality of context objects, each of said plurality of context objects to store values for one or more of the input parameters to said first function; and accessing said values from said default context object.

53. The method of claim 47, further comprising:

applying a first method from a second execution object, said second execution object also identifying said first metadata object, said first method of said second execution object causing the acts of,
accessing the data describing each input parameter of said first function from said first metadata object;
attempting to match values associated with said second execution object to each input parameter of said first function as described by the data in said first metadata object; and
determining one or more values are missing for at least certain input parameters of said first function.

54. A method comprising:

applying a first method from a first execution object, said first execution object identifying a first of a plurality of metadata objects, each of said plurality of metadata objects identifying a different function, each of said functions having input and output parameters, wherein one or more parameters for different functions are the same, the parameters for the different functions collectively defining a set of parameter kinds, each parameter kind in said set of parameter kinds being assigned a unique key, each of said plurality of metadata objects storing the unique keys assigned the input and output parameters of the function they identify, said first method causing the acts of,

accessing the key for each input parameter stored in said first metadata object; attempting to match parameter values associated with said first execution object to each of the accessed keys; and determining parameter values are missing for a set including at least one of the accessed keys.

55. The method of claim 54, wherein each parameter of said functions is of one of a plurality of data types each supporting a range of values, wherein different data is categorized irrespective of data type, and wherein each category of data defines one of the set of parameter kinds.
56. The method of claim 54, wherein said first method further causes the acts of: locating a set of one or more of said plurality of metadata objects that collectively store each of the set of keys; and executing the functions corresponding to the set of metadata objects to acquire said set of missing parameter values as outputs of the functions; associating the acquired parameter values with the first execution object; and executing the function identified by the first metadata object using the parameter values associated with the first execution object as input parameters.
57. The method of claim 54, wherein said attempting further includes: accessing a structure in said first execution object, said structure in said first execution object to store values for one or more said set of input parameters to the function identified by the first metadata object.
58. The method of claim 54, wherein said attempting further includes: accessing a structure in a manager object identified by a structure in said first execution object, said structure in said manager object identifying a default one of a

plurality of context objects, each of said plurality of context objects to store values for one or more of the input parameters to said first function; and accessing one or more of said values from said default context object.

59. The method of claim 54, further comprising:
 - applying a first method from a second execution object, said second execution object also identifying said first metadata object, said first method of said second execution object causing the acts of,
 - accessing the key for each input parameter stored in said first metadata object; attempting to match values associated with said second execution object to each of the accessed keys; and
 - determining parameter values are missing for a set including at least one of the accessed keys.
60. The method of claim 54, wherein said attempting further includes:
 - accessing from a first structure in said first execution object a value for a first input parameter to said function identified by said first metadata object;
 - accessing, from a first of a set of context objects that was passed, a value for a second input parameter to said first function, said set of context objects being stored in a first structure of a manager object, each of said set of context objects to store values for one or more of the input parameters to said first function, said manager object identifying one of said set of context objects as a default context object;
 - and
 - accessing from said default context object a value for a third input parameter to said first function.
61. A method comprising:

receiving a request to locate a function that provides a particular parameter kind as an output;

locating a metadata object having stored therein data identifying said particular parameter kind, said metadata object identifying a function and storing said data to indicate the particular parameter kind is an output parameter of said function; and

providing an execution object that identifies said metadata object, wherein said execution object includes,

a structure to identify values for a set of one or more input parameters to said function, and

a method, which when applied, causes said function to be invoked using the values identified by said structure as input parameters.

62. The method of claim 61, wherein:
said function has a plurality of parameters, said metadata object stores data identifying each kind of said plurality of parameters.
63. The method of claim 62, wherein each parameter of said function is of one of a plurality of data types each supporting a range of values, wherein different data is categorized irrespective of data type, and wherein each category of data defines one of the parameter kinds.
64. The method of claim 61, wherein:
said metadata object includes an action method, which when applied by a processor, causes said processor to invoke said function; and
said method in said execution object, when applied, causes said action method to be applied.
65. A method comprising:

receiving a request to locate a function that provides a particular output parameter, wherein each of a plurality of metadata objects identify a different function having one or more output parameters, said output parameters for the different functions collectively defining a set of parameter kinds, each parameter kind in said set being assigned a unique key, each of said plurality of metadata objects storing the unique keys assigned the output parameters of the function they identify;

locating a first of said plurality of metadata objects that stores the unique key for the particular output parameter; and

creating an execution object that identifies said first metadata object, wherein said execution object includes,

a structure to identify values for a set of one or more input parameters to said function identified by said first metadata object, and

a method, which when applied, causes said function identified by said first metadata object to be invoked using the values identified by said structure as input parameters.

66. The method of claim 65, wherein each parameter of said functions is of one of a plurality of data types each supporting a range of values, wherein different data is categorized irrespective of data type, and wherein each category of data defines one of the parameter kinds.

67. The method of claim 65, further comprising:

applying said method from said execution object, wherein both said input and output parameters for the different functions collectively define said set of parameter kinds, each of said plurality of metadata objects storing the unique keys assigned the input and output parameters of the function they identify, said method from said execution object causing the acts of,

accessing the key for each input parameter stored in said first metadata object;

attempting to match parameter values associated with said execution object to each of the accessed keys; and executing the function identified by said first metadata object using the parameter values associated with the execution object as input parameters.

68. The method of claim 65, wherein said attempting further causes the acts of: determining parameter values are missing for a set including at least one of the accessed keys.

locating a set of one or more of said plurality of metadata objects that collectively store each of the set of keys for output parameters; and executing the functions corresponding to the plurality of metadata objects to acquire said set of missing parameter values as outputs of the functions; and associating the acquired parameter values with first execution object.

69. A method comprising:

applying a first method from a first execution object, said first execution object identifying a first of a plurality of metadata objects, each of said plurality of metadata objects identifying a different function, each of said functions having input and output parameters, wherein one or more parameters for different functions are the same, said parameters for the different functions collectively defining a set of parameter kinds, each parameter kind in said set of parameter kinds being assigned a unique key, each of said plurality of metadata objects storing the unique keys assigned the input and output parameters of the function they identify, said first method causing the acts of,

accessing the key for each input parameter stored in said first metadata object;

associating with said first execution object a value stored as part of a first of a set of context objects that was passed, said value stored for use as a first input parameter, said set of context objects being stored in a first structure of a manager object, each of said set of context objects to store values for one or more of the input parameters to said first function; and

executing said first function using the parameter values associated with the first execution object as input parameters.

70. The method of claim 69, wherein said first method further causes the acts of: associating with said first execution object a value stored as part of one of said set of context objects identified by said manager object as a default context object, said value stored for a second input parameter to said first function.